Appl. No.: 09/740,419
Amdt. dated January 23, 2004
Reply to Office action of Novemb r 17, 2003

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

A1

1. (Original) A computer system, comprising:

a processor;

a system memory coupled to said processor; and

an input device coupled to said processor;

said processor having a branch predictor, said branch predictor includes a multi-bank prediction array, each of said banks comprising a single-ported memory device;

said branch predictor also including bank control logic coupled to said prediction array to ensure that two accesses to said prediction array in the same cycle do not conflict.

2. (Original) The computer system of claim 1 wherein said processor further includes fetch logic that fetches two slots of instructions in one cycle.

3. (Original) The computer system of claim 1 wherein said branch predictor further includes a multiplexer coupled to each of said single-ported banks and controlled by said bank control logic.

4. (Original) The computer system of claim 1 wherein said branch predictor determines an index value based on a conditional branch instruction and uses said index value to retrieve a prediction from said prediction array.

5. (Original) The computer system of claim 4 wherein each of said banks has an identifier and said bank control logic determines a bank identifier for a conditional branch instruction that is different than the bank identifier determined

A1

for a conditional branch instruction that was last used to access said prediction array.

6. (Original) The computer system of claim 4 wherein said bank control logic selects two bits from said index value to be a bank number.

7. (Original) The computer system of claim 4 wherein said bank control logic selects two bits from said index value to be a bank number if the value of said two bits does not equal a bank number determined for a conditional branch instruction that was last used to access said prediction array.

8. (Original) The computer system of claim 7 wherein, if said two bits equals said bank number determined for a conditional branch instruction that was last used to access said prediction array, then said bank control logic changes the values of said two bits and uses the changed value as a bank number.

9. (Original) The computer system of claim 8 wherein said bank control logic changes the value of said two bits by incrementing the value of said two bits.

10. (Original) The computer system of claim 3 wherein said branch predictor further includes a pair of 4-to-1 multiplexers that receive output signals each of said single-ported banks, said pair of multiplexers are controlled by said bank control logic.

11. (Original) The computer system of claim 1 wherein said processor further includes fetch logic that fetches at least two slots of instructions in one cycle.

12. (Currently amended) A processor, comprising:

a multi-bank branch prediction array, each of said banks comprising a single-ported memory device; and

bank control logic coupled to said prediction array to ensure that two branch prediction accesses to said prediction array in the same cycle do not conflict.

A1

13. (Original) The processor of claim 12 wherein said processor further includes fetch logic that fetches two slots of instructions in one cycle.

14. (Original) The processor of claim 12 wherein said branch predictor further includes a multiplexer coupled to each of said single-ported banks and controlled by said bank control logic.

15. (Original) The processor of claim 12 wherein said branch predictor determines an index value based on a conditional branch instruction and uses said index value to retrieve a prediction from said prediction array.

16. (Original) The processor of claim 15 wherein each of said banks has an identifier and said bank control logic determines a bank identifier for a conditional branch instruction that is different than the bank identifier determined for a conditional branch instruction that was last used to access said prediction array.

17. (Original) The processor of claim 15 wherein said bank control logic selects two bits from said index value to be a bank identifier.

18. (Original) The processor of claim 15 wherein said bank control logic selects two bits from said index value to be a bank number if the value of said two bits does not equal a bank identifier determined for a conditional branch instruction that was last used to access said prediction array.

19. (Original) The processor of claim 18 wherein, if said two bits equals said bank identifier determined for a conditional branch instruction that was last used

to access said prediction array, then said bank control logic changes the values of said two bits and uses the changed value as a bank identifier.

A1

20. (Original) The processor of claim 19 wherein said bank control logic changes the value of said two bits by incrementing the value of said two bits.

21. (Original) The processor of claim 14 wherein said branch predictor further includes a pair of 4-to-1 multiplexers that receive output signals each of said single-ported banks, said pair of multiplexers are controlled by said bank control logic.

22. (Original) A method of avoiding bank conflicts in a multi-bank prediction array in a processor, comprising:

generating an index value from a conditional branch instruction address;

selecting two bits from said index value;

comparing the value of said two bits with a previous bank number determined for a conditional branch instruction previously used to a access said prediction array;

using the value of said two bits as a current bank number if said value of said bits differs from said previous bank number;

if said value of said two bits equals said previous bank number, changing the value of said two bits to be the current bank number; and

using said current bank number to access the corresponding bank in said prediction array to retrieve a prediction.

23. (Original) The method of claim 22 wherein changing the value of said two bits comprises incrementing the value of said two bits.